

# Programmation coté serveur : PHP

## Atelier 3

Un atelier PHP étape par étape pour créer les services suivants en utilisant MySQL et PDO :

### 1. Inscription

#### a. Créer une base de données MySQL :

```
CREATE DATABASE user_management;
USE user_management;

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    email VARCHAR(100) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL
);
```

#### b. Configuration PDO pour la connexion à la base de données :

```
<?php
// db.php
$host = '127.0.0.1';
$db = 'user_management';
$user = 'root';
$pass = 'password'; // Remplacez par votre mot de passe MySQL
$charset = 'utf8mb4';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";
$options = [
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES => false,
];

try {
    $pdo = new PDO($dsn, $user, $pass, $options);
} catch (PDOException $e) {
    throw new PDOException($e->getMessage(), (int)$e->getCode());
}
?>
```

### c. Formulaire d'inscription :

```
html
<!-- register.html -->
<form method="POST" action="register.php">
    <input type="text" name="username" placeholder="Nom d'utilisateur"
required>
    <input type="email" name="email" placeholder="Email" required>
    <input type="password" name="password" placeholder="Mot de passe"
required>
    <button type="submit">S'inscrire</button>
</form>
```

### d. Script d'inscription (hash du mot de passe) :

```
<?php
// register.php
require 'db.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT);

    $stmt = $pdo->prepare('INSERT INTO users (username, email, password)
VALUES (?, ?, ?)');
    $stmt->execute([$username, $email, $password]);

    echo 'Inscription réussie';
}
?>
```

## 2. Connexion

### a. Formulaire de connexion :

```
<!-- login.html -->
<form method="POST" action="login.php">
    <input type="email" name="email" placeholder="Email" required>
    <input type="password" name="password" placeholder="Mot de passe"
required>
    <button type="submit">Se connecter</button>
</form>
```

### b. Script de connexion (vérification du mot de passe) :

```
<?php
// login.php
session_start();
require 'db.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $email = $_POST['email'];
    $password = $_POST['password'];

    $stmt = $pdo->prepare('SELECT * FROM users WHERE email = ?');
    $stmt->execute([$email]);
    $user = $stmt->fetch();
```

```

        if ($user && password_verify($password, $user['password'])) {
            $_SESSION['user_id'] = $user['id'];
            echo 'Connexion réussie';
        } else {
            echo 'Email ou mot de passe incorrect';
        }
    }
}
?>

```

### 3. Déconnexion

#### a. Script de déconnexion :

```

<?php
// logout.php
session_start();
session_unset();
session_destroy();

echo 'Déconnexion réussie';
?>

```

### 4. Gestion de session

#### a. Vérifier si l'utilisateur est connecté :

```

<?php
// session_check.php
session_start();

if (!isset($_SESSION['user_id'])) {
    echo 'Vous n\'êtes pas connecté';
    exit;
}

echo 'Vous êtes connecté en tant qu\'utilisateur ID : ' .
$_SESSION['user_id'];
?>

```

### 5. Connexion avec Google et Facebook

Pour ajouter l'authentification avec Google et Facebook, vous devrez utiliser des SDK spécifiques pour chaque service.

#### a. Google OAuth :

- Vous devrez configurer l'API Google pour obtenir des identifiants OAuth.
- Utilisez la bibliothèque PHP `google/apiclient` via Composer.

```
composer require google/apiclient
```

#### b. Facebook OAuth :

- Utilisez le SDK Facebook via Composer.

```
composer require facebook/graph-sdk
```

L'intégration de ces services requiert des configurations spécifiques et l'utilisation d'API pour la gestion des jetons OAuth. Une fois configuré, ces connexions OAuth peuvent être combinées avec la gestion des utilisateurs dans votre base de données.